

CS 150 Lab

Introduction to Visual Studio

The purpose of today's lab session is to familiarize you with the Computer Science server 'Turing' where you will be submitting all your assignments and lab projects. We will also use Visual Studio to compile and run basic C++ programs.

Lab 1.1

We will work through the first problem together. We are going to connect to the Computer Science server, and write a program to display the message 'Hello World!'.

Logging on:

1) Log on to the computer using your PU net ID and password. Your PUnet ID is the first four letters of your last name and the last four letters of your id number. Your password is your seven digit ID number.

Connecting to Turing:

Turing is the computer science server where all of your work will be stored. Anything you save locally on the machines in the lab will be deleted once you log off, so it's important to save your files on Turing in order not to lose your documents.

Before you can save anything to the server you will need to connect to it.

2) Connect to (Turing):

- a. **Open 'My Network Places' on the desktop.**
- b. **Double click on 'Add Network Place'.**
- c. **In the text box type: [\\turing.cs.pacificu.edu\students](http://turing.cs.pacificu.edu/students). Click 'next'.**
- d. **You will be prompted to type a name and password. Type your PU net ID as your name and your student ID as your password. Click 'next'.**
- e. **You will be prompted to name this network place. Type 'Turing' in the text box. Click 'finish'.**

This will open up the Students folder. This folder contains a folder called "cs150-01 Fall07", which should contain a folder with the same name as your PU net ID. This is your personal folder that you can use to store anything related to CS150. No other student has access to this folder, but faculty members do. Do not store any illegal material on Turing.

Changing your Password:

To change your password, write down on the piece of paper I give you:

- Your name
- Your student ID
- Your new password

Be sure to write the new password clearly!!

Hand them to me by the end of class and the password will be changed by next week.

Starting Visual Studio:

Visual Studio is the integrated development environment (IDE) that you will use to develop your C++ programs. The software will be available for you to install on your home machines next week.

3) Open Visual Studio by clicking the Start menu, selecting Programs, then Developer Tools, then Microsoft Visual Studio 2005 then Microsoft Visual Studio 2005.

Visual Studio will start with no projects open.

Today, we are going to create a new C++ program and compile and run it. You are also going to copy your work to the “CS150-01 Lab” drop box, which is where you will be submitting all your future lab projects.

Creating a new project:

To create a new project, **select File -> New -> Project from the menus.**

This opens a dialog box with several options. The window on the left lists the project types, while the window on the right lists the templates. All the projects that we will create this semester will be Win32 Console Projects

4) Expand *Visual C++* from the left window and click on the *General* option. Select the *Empty Project* from the right window.

You will select these options for all your projects this semester.

5) In the Name textbox, enter a name for your project. For this project, call it ‘01HelloWorldxxxxxxx’ replacing the x’s with your PUNet ID. Your PUNet ID is the login you use for campus machines. It’s the first four letters of your last name, followed by the last four digits of your id number.

6) Note the location where the project will be saved. You can change the location so that the project is saved on the desktop. Expand the drop down arrow and select the desktop.

7) Click the OK button.

Congratulations! You have created your first C++ project.

The left or right hand side of the window lists all the files that are in your project. It will contain three folders (Header, Resource, and Source Files). They are all currently empty. We need to add a file to write your program in.

8) Right click on *Source Files*, select *Add*, then *New Item*. In the dialog box, expand the *Visual C++* item, select *Code*, and from the right hand side select *C++ File*. Enter a name for the file, the name should be: *main.cpp*. Click *Add*.

Writing the Program:

It is time to write your first C++ program. Type the following code into main.cpp. You don't need to worry about the color-coding, Visual Studio will do that for you. Also, you do not need to type the line numbers.

```
1//*****
2 // File name:  main.cpp
3 // Author:    John Smith
4 // Date:      08/28/2007
5 // Class:     CS150
6 // Assignment: Hello World Lab Program
7 // Purpose:   Displays the message "Hello World" to the
8 //            screen.
9//*****
10
11 #include <iostream>
12
13 using namespace std;
14
15 int main()
16 {
17     cout << "Hello World!" << endl;
18     return 0;
19 }
```

Examining the program:

Lines 1 through 9 in green at the top are just comments. You can ignore these for now. When you start writing your own programs you should write your name and the date as comments in your program. You should also write down the functionality of your program in a paragraph or two.

Line 11 includes the `iostream` (input/output stream) library required to display messages to the screen. It's also required when reading input from the user.

Line 13 specifies the namespace, which is required for using the keyword `cout` to display messages to the screen.

Line 15 declares the main function that is required by every C++ program.

Line 17 sends the message "hello world!" to the screen. `cout <<` indicates that we want to write to the standard output (the screen). The text between double quotes (") is what is written on the screen after running the program. The `endl` at the end of the line moves the output cursor to a new line.

Line 18 contains `'return 0'` which just tells the program to exit normally.

Building the program:

Before you can run your program and see it in action you will need to build it. Building the program checks it for syntax errors and translates the C++ code into machine code that is understood by the computer.

13) To build the program, click on 'Build' from the menu, then 'Build Solution'.

The program will then compile and link. The output window below the source window will display messages indicating what is going on. You can expand this window or scroll through so that you are able to read all the messages.

If you have typed everything correctly, then you should have 0 errors and 0 warnings. If you do get errors, then check your code and make sure that it matches mine. Let me know if it still doesn't work.

Q 1.1. What happens if we type `cot` instead of `cout`? Try it now, and build and run your program again.

A. _____

Running the program:

Now that we have built the program, we can run it. Running the program will execute the C++ code that we have just written.

14) To run the program, click ‘Debug’ from the menu then click ‘Start Without Debugging’. You can also hit ctrl-F5.

What happens next?

A new window should open with the text ‘Hello World!’ displayed. There will also be a line that says ‘Press any key to continue’. Pressing any key will close the window.

Congratulations! You have run your first C++ program.

15) Press any key to close the window.

Q 1.2. What happens if we delete `<< endl` from line 18? Try it now, and build and run your program again.

A. _____

Saving your projects onto Turing

Before you log off of the local machines, you should save your work to your folder on Turing so that it is not lost. You do this by opening quitting Visual Studio and going to the desktop. Open your folder on Turing, and drag the project folder to your folder (the one with your PU Net ID) on Turing. You must drag the entire project folder to save everything correctly.

Show your completed submitted programs to the TA or professor

Once you have completed the program and submitted it to Turing, please show it to the TA or the professor so that it can be graded.

Lab 1.2

Once you have completed all of the above, then create a new project following the same steps as above, and copy out Program 1-1 from page 13 of your textbook. I have copied the program below. Name this project "02Payxxxxxxx" replacing the x's with your PUNet ID.

```
// This program calculates the user's pay.
#include <iostream>
using namespace std;

int main()
{
    double hours, rate, pay;

    // Get the number of hours worked.
    cout << "How many hours did you work? ";
    cin >> hours;

    // Get the hourly pay rate.
    cout << "How much do you get paid per hour? ";
    cin >> rate;

    // Calculate the pay.
    pay = hours * rate;

    // Display the pay.
    cout << "You have earned $" << pay << endl;
    return 0;
}
```

Build and run your program.

Q1.3. What does the program do? Can you identify what each statement does? Make a note of these here.

A. _____

Show your completed work to the TA or instructor.

Submitting your finished project:

Once you have completed your lab projects or assignment, you will need to submit it for grading. You will submit your projects to the 'CS150-01 Lab' folder on. You may only submit your projects once, so make sure that they are in their final form. Also, make sure that you submit the correct project.

24) Exit Visual Studio.

25) From the desktop, double click on 'My Network Places'. Double click 'Turing, then double click your folder. Create a folder called '01xxxxxxx', where you replace the x's with your PU Net ID. Place your two projects '01HelloWorldxxxxxxx' and '02Payxxxxxxx' into this folder.

Don't close that window!

26) Double click on 'My Network Places', and double click on 'Turing. This should open a new window.

27) Locate the folder called 'CS150-01 Lab'.

28) Now go back to the other window and drag the folder called '01xxxxxxx' and place it in the folder called 'CS150-01 Lab'.

This is how you will hand in all of your lab programming assignments.

Lab1.3 Optional Challenge:

Once you have followed all of the above instructions and submitted your lab project, modify your project so that it displays a message asking for a name. Once the user types in the name, the program should display "Hello" followed by the name typed in by the user.

For example, if the user types in 'Bob', the program output should be:

```
What is your name? Bob  
Hello Bob!
```